

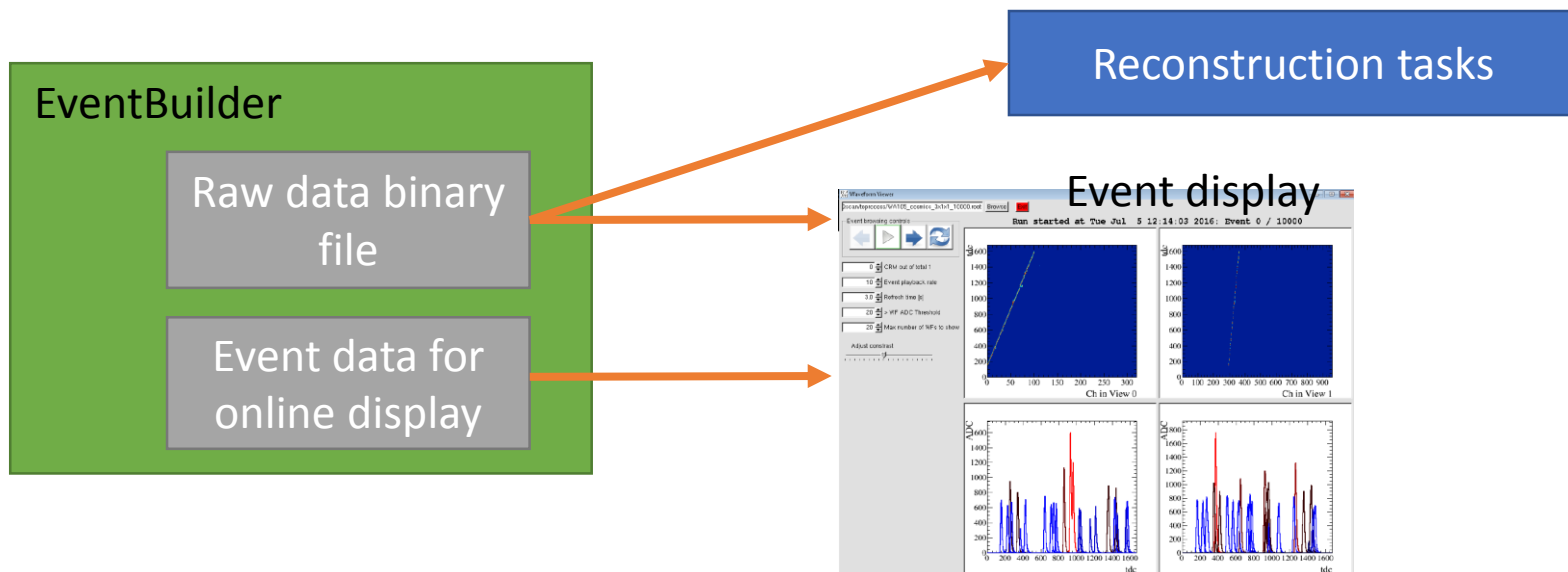
~~WA105~~ 

QSCAN interface for the 3x1x1 raw data

SB Meeting 12.10.2016

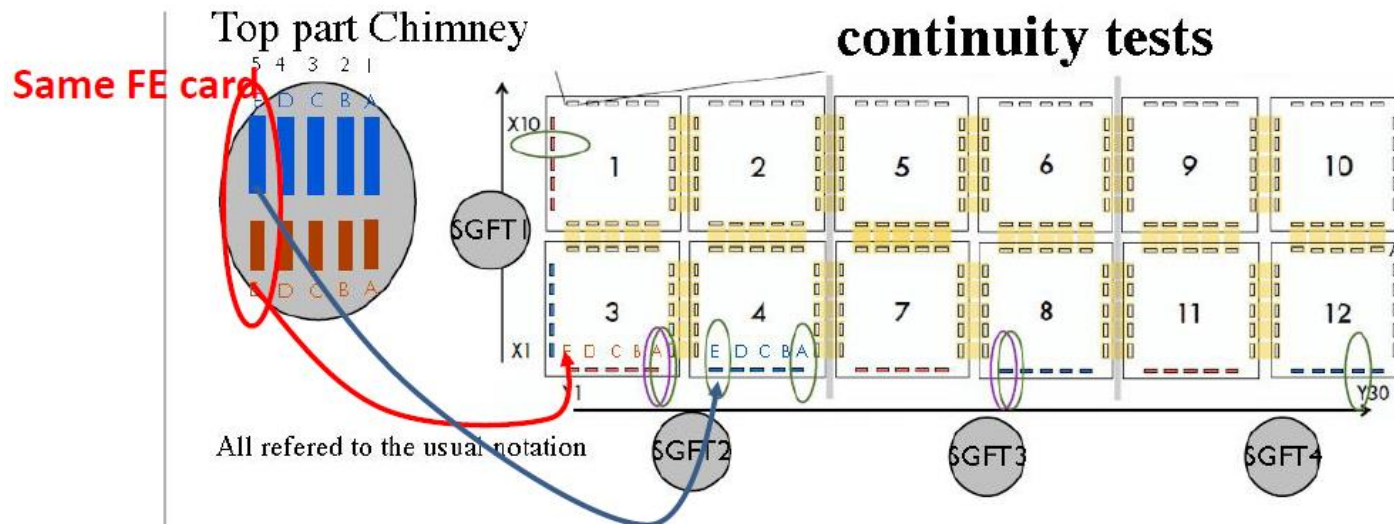
Overview

- Need to decode events from raw data files produced by DAQ
- Integrate into existing QSCAN structure in a transparent way (i.e., the software or user should not care if one runs on MC or raw data files)
 - Event display
 - Reconstruction / analysis
- This structure has been already developed and tested last summer, but ch mapping mechanism between DAQ and CRP views needs to be generalized to reflect the fact that one card does not read a continuous block of 64 channels



Mapping DAQ channels to “view” channels

DAQ channel mapping on the CRP



Event data from EB in the file is **1280 ch x 1667 samples**.

Ordering of 1280 channel samples:

Crate 0

Card 0

Ch 0...63 → Counting from top-to-bottom

Card 1

Ch 0...63

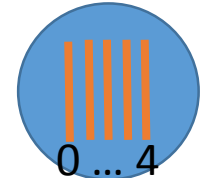
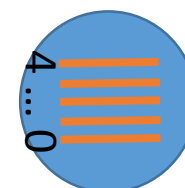
...

Crate numbering:

0 == SGFT1 (with 3m strips)

1 == SGFT2...

Card order in chimney / crate:



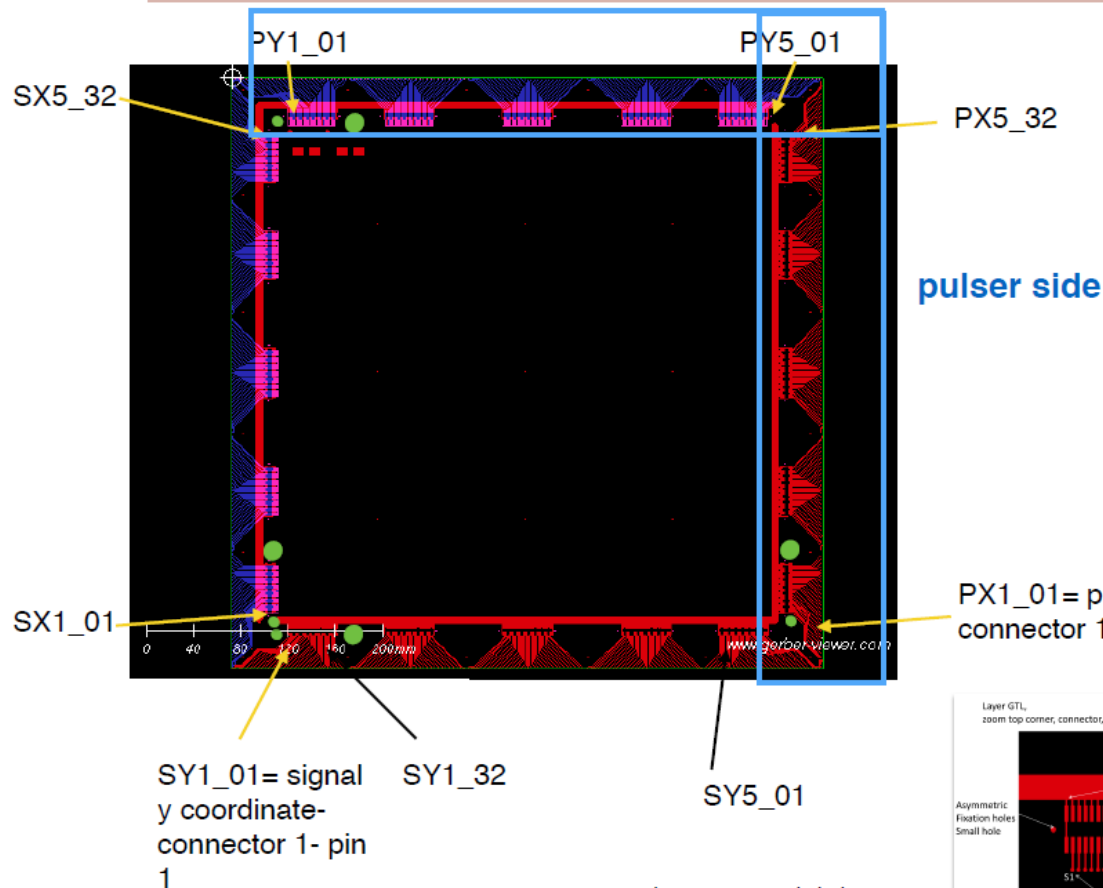
Channel numbering

The channel numbering per connector is fixed by the orientation of the connector key

ETH

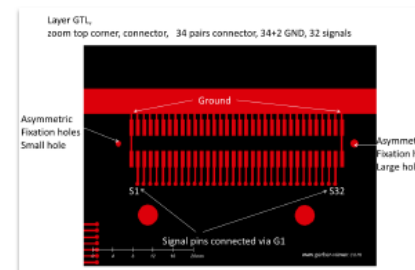
Layout

WA105



From 24/03/2016
3x1 meeting

connectors have asymmetric holes to not make mistakes.
Ch 1 always starts on the small hole.



Channel numbering

is propagated to flange...

Warm flange with

VHDCI connectors to uTCA DAQ

Low voltages
SUB-D 9

VHDCI A

VHDCI B

0→31

32→63

Rulser
SMA

Slow control
SUB-D 9

Molex to uTCA crate (32 channels)

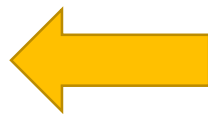
Pin		Pin	
1	S1_P	35	S1_N
2	S2_P	36	S2_N
3	S3_P	37	S3_N
....			
16	S16_P	50	S16_N
17	GND	51	GND
18	GND	52	GND
19	S17_P	53	S17_N
.....			
33	S31_P	67	S31_N
34	S32_P	68	S32_N

From 02/06/2016
3x1 meeting

The channel numbering on connectors is given by the orientations of the connector key and followed by the DAQ (1 card reads 2x32 channels)
Then we have to take care of is which part of anode each connector maps to

Mapping DAQ channels to “view” channels

View X: 320 ch (0...319) (SGFT1)
View Y: 960 ch (0...959) (SGFT2-4)



This is the representation we need to built for recon/analysis

Each card reads a block of 64 ch, but it is split between two neighboring anodes at the level of cable-anode connections

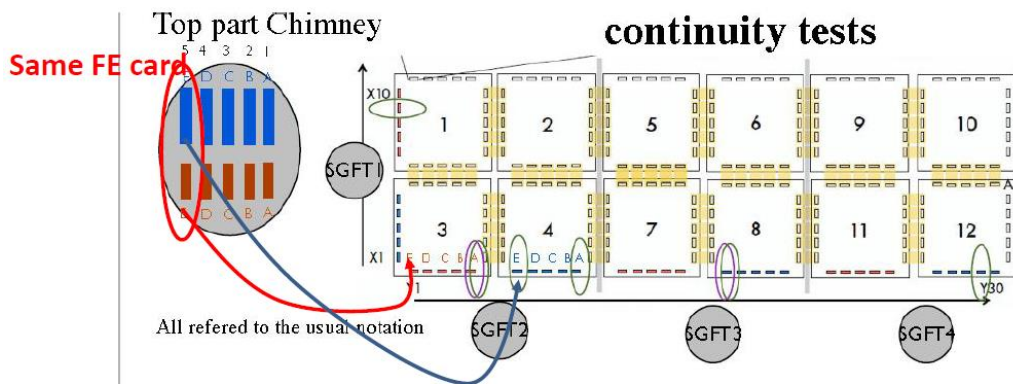
E.g.,

View X

Crate	Card	Ch	View Ch
0	0	31	31
0	0	32	160
0	4	31	159
0	4	32	288

View Y

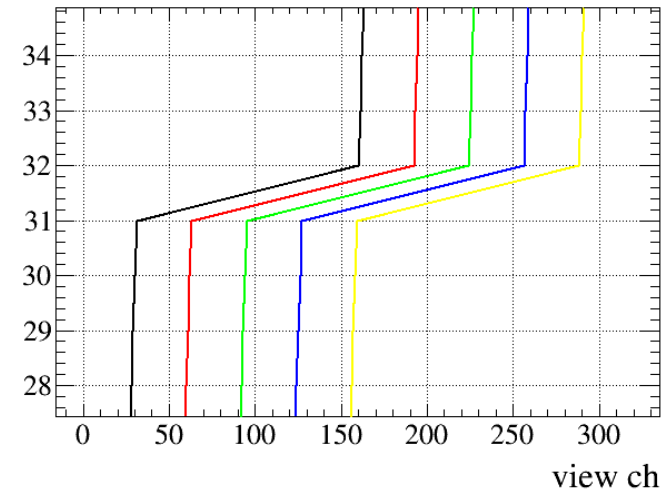
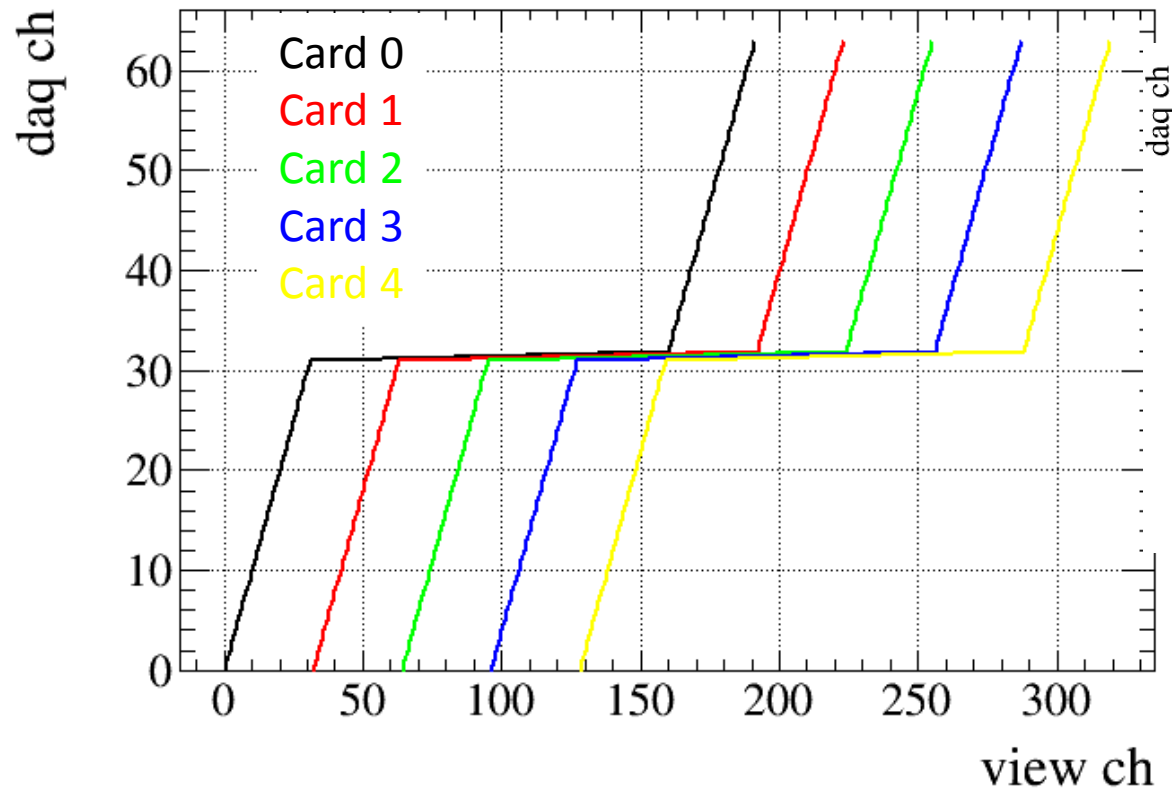
Crate	Card	Ch	View Ch
1	0	31	191
1	0	32	0
1	4	31	319
1	4	32	128
2	0	0	480



With the start of counting is as shown

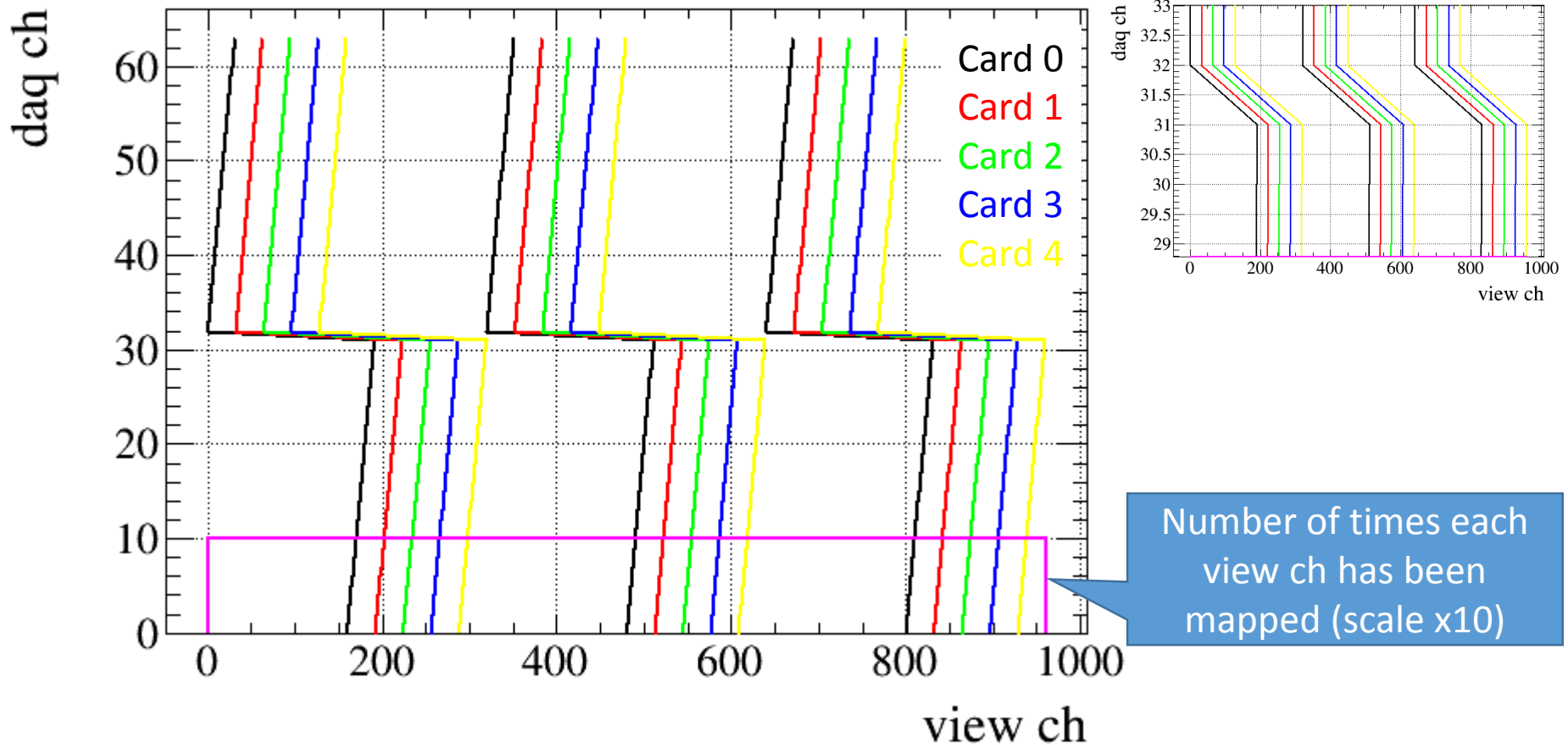
- In x the 0-31 (top connector) channels map to the first half, while the 32-63 are mapped to second half
- In y 0-31 (top connector) maps to second half of 1m² sub-crp block while 32-63 to the first

Graph of mapping function: view X



Unique mapping → the lines should never cross

Graph of mapping function: view Y



Unique mapping → the lines should never cross

Added library

Index of /WA105Soft/daq/src

Files shown: 6

Directory revision: [366](#) (of [369](#))

Sticky Revision:

[File](#) ▲

[Parent Directory](#)

[ChMapInterface.cc](#)

[ChannelMap.cc](#)

[EventDecoder.cc](#)

[EventEncoder.cc](#)

[HuffDataCompressor.cc](#)

[dlardaq.cc](#)

DAQ channel mapping for 3x1x1 is parametrized in ChannelMap311 object

Decoder for raw event data

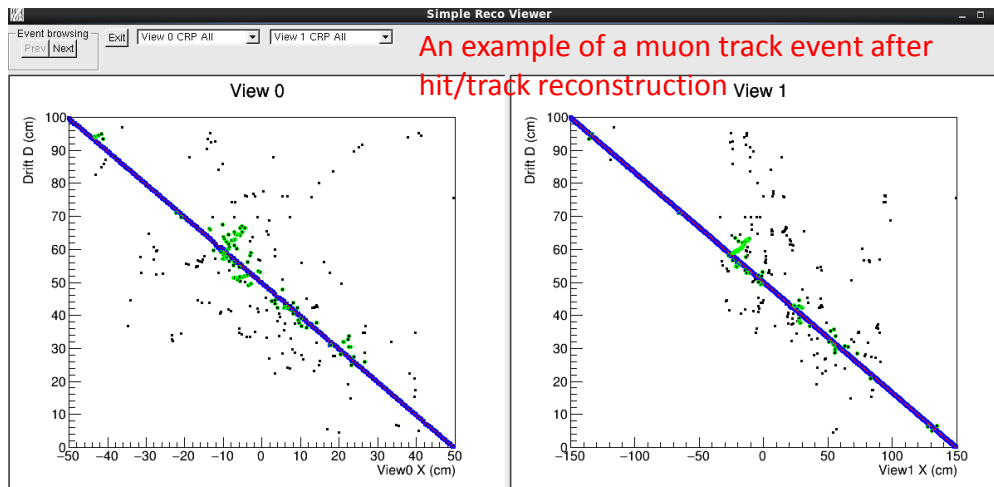
Detector coordinate system

Once the DAQ channels are mapped to view channels the latter are used to convert to the detector coordinate system at the level of hit reconstruction

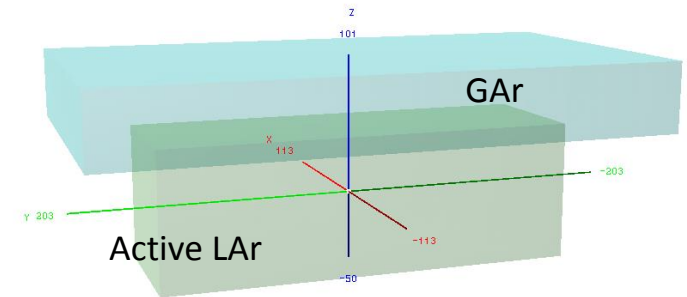
The origin of this coordinate system is in the middle of the active volume

ViewXCh 0 \rightarrow -50 cm in detector coordinates

ViewYCh 0 \rightarrow -150 cm in detector coordinates



3D view of MC geometry with inner volume and gas layer shown



This origin is fixed and ensures that the data and MC use the same coordinate system. So it should be used as reference in recon / analyses

Header information from DAQ to QSCAN

Run header

```
// structure to hold decoded run header
typedef struct runheader_t
{
    uint32_t run_num;
    uint8_t  run_flags;
} runheader_t;
```

Basically contains run number
+ byte reserved for flags

Event header

```
// structure to hold decoded event header
typedef struct evheader_t
{
    trigger_t trig_info; // trigger info
    uint8_t dq_flag;     // data quality flag
    uint32_t ev_num;     // event number
    uint32_t ev_size;    // size of event in bytes
} evheader_t;
```

Contains trigger information
Data quality bits + compression flag
Event number
Event size (useful for compressed events)

```
typedef struct trigger_t
{
    uint8_t type;
    uint32_t num;
    struct timespec ts; // { time_t ts.tv_sec, long tv_nsec }
} trigger_t;
```

The trigger part of event header contains
Trigger type flag
Trigger number
Trigger time from WR stored in unix *timespec* structure

The information from these items will be mapped to the RunHeader/EventHeader objects in QSCAN to be available at the level of recon & analysis

Summary

- View naming convention:
 - View 0 == View X is read out by SGFT1 (320ch 3 m strips)
 - View 1 == View Y is read out SGFT2-4 (960 ch 1 m strips)
- Coordinative axis origin: middle of the 3x1x1 active volume
- Module to decode raw event data is ready
 - Both Huffman compressed and uncompressed events are supported
- The DAQ channel mapping in QSCAN for reconstruction and analysis has been prepared